

Online Research @ Cardiff

This is an Open Access document downloaded from ORCA, Cardiff University's institutional repository: <https://orca.cardiff.ac.uk/id/eprint/109428/>

This is the author's version of a work that was submitted to / accepted for publication.

Citation for final published version:

Thomas, Lee ORCID: <https://orcid.org/0000-0002-4602-5102>, Long, Chao ORCID: <https://orcid.org/0000-0002-5348-8404>, Burnap, Peter ORCID: <https://orcid.org/0000-0003-0396-633X>, Wu, Jianzhong ORCID: <https://orcid.org/0000-0001-7928-3602> and Jenkins, Nicholas ORCID: <https://orcid.org/0000-0003-3082-6260> 2017. Automation of the supplier role in the GB power system using blockchain based smart contracts. CIRED - Open Access Proceedings Journal 2017 (1) , pp. 2619-2623. 10.1049/oap-cired.2017.0987 file

Publishers page: <http://digital-library.theiet.org/content/journals...>
<<http://digital-library.theiet.org/content/journals/10.1049/oap-cired.2017.0987>>

Please note:

Changes made as a result of publishing processes such as copy-editing, formatting and page numbers may not be reflected in this version. For the definitive version of this publication, please refer to the published source. You are advised to consult the publisher's version if you wish to cite this paper.

This version is being made available in accordance with publisher policies.

See

<http://orca.cf.ac.uk/policies.html> for usage policies. Copyright and moral rights for publications made available in ORCA are retained by the copyright holders.



Automation of the supplier role in the GB power system using blockchain-based smart contracts

Lee Thomas¹✉, Chao Long¹, Pete Burnap², Jianzhong Wu¹, Nick Jenkins¹

¹School of Engineering, Cardiff University, Wales, UK

²School of Computer Science and Informatics, Cardiff University, Wales, UK

✉ E-mail: ThomasL62@cf.ac.uk

Abstract: An electricity supply smart contract was developed and demonstrated to perform pre-time-of-use price negotiation between demand and generation and post-time-of-use settlement and payment. The smart contract was demonstrated with 1000 loads/generators with usages simulated using lognormal probability distributions. It combines payment of deposit, negotiation of price based on estimates, settlement based on actual usage and enactment of payments using crypto-currency. The settlement procedure rewards customers that adjusted to balance the system. The smart contract was written in the solidity programming language and implemented with a simulated Ethereum blockchain using testrpc and go-ethereum. In the example test case, a price was agreed, settled and payment enacted.

1 Introduction

Automation of the energy supply role in the GB power system will become possible due to the increasing feasibility of smart contracts and the increase in digital metering across the power system. On the GB system, energy suppliers act as the interface between consumers and generators. They are free to negotiate prices with generators and compete for customers with other suppliers. The bulk of the demand–supply balance is achieved through these negotiations, with the remaining short-term imbalance managed by the system operator through the balancing mechanism. Through demonstration of automated negotiation, settlement and payment, with reward for system balancing, this work shows the potential for blockchain-based smart contracts to undertake the supplier role.

The intent of the supplier's role, established during the market liberalisation of the 1980s and 1990s, is that those suppliers offering the most competitive offers (in terms of price and/or generation mix) will tend to survive. However, an investigation found that a relatively low proportion of consumers change suppliers or negotiate (~70% of domestic customers stay on default tariffs) [1]. There is, therefore, desire to improve the system. The advent of implementable smart contracts may give governments and regulators opportunity to reduce overall system costs and ensure competitiveness.

Smart contracts are rules for information exchange that have a common mechanism for definition and enactment. They were described by Szabo in 1994 [2], who later envisaged a mutually trusted virtual computer upon which smart contracts could run [3]. Implementations of such virtual computers using modern cryptography are commonly associated with the label 'blockchain technology'.

Blockchains are data structures (blocks) with cryptographic links from the present state to the preceding state. When changes are made to a block in accordance with a set of rules, a new block is created. Each new block must also contain a cryptographic hash of the previous block. This creates a continuously growing chain of blocks. In many public blockchains, the latest block is agreed upon using proof of work or proof of stake consensus algorithms.

The growth of blockchain technology for cryptocurrency is well documented [4]. Recent advances go further in allowing transactions to define smart contracts. One example, Ethereum, includes a blockchain-based 256 bit virtual computer [5] – this allows flexibility in how smart contracts are defined.

There is worldwide research and implementation activity, including many start-up companies [6–13], in the area of blockchain-based smart contracts for energy systems. Aitzhan and Svetinovic describe a token-based peer-to-peer system for anonymous negotiation of energy prices [14] and Horta *et al.* describe work underway on a blockchain-based distributed energy market test system [15]. LO3Energy's Brooklyn microgrid project [16] has demonstrated the interaction of electricity metering with a blockchain-based smart contract and Ponton claimed the first blockchain-based energy trade in Europe [8]. SolarCoin is a token for reward of solar electricity generation, the token is created when meters (registered with the SolarCoin Foundation) record solar electricity generation [17].

2 Method

2.1 Example supplier smart contract

A simple supplier smart contract was implemented on the Ethereum blockchain platform, using a private test net. It shows how a set of rules for a price negotiation and settlement, in which all connected users (demand and generation) have half hourly metering, are defined and enacted. The contract process is shown in Figs. 1 and 2.

Users (load/generator connections) are classified as critical or non-critical. Critical users are those who must not be disconnected and play no role in price negotiation. Non-critical users submit offers (either generation or demand) in Wh with minimum price threshold for the next hour's usage and only operate if their offer is accepted. Two mappings of demand offers and generation offers are created, both referenced by offer price.

All critical users submit a predicted usage in Wh for the hour ahead to the contract. Losses are estimated by the network operator (simplified to a single entity). A system wide demand/generation imbalance figure is calculated by the contract. If there is excess demand, generation offers are accepted (in order of lowest offer price to highest) until there is excess generation. Conversely, if there is excess generation, demand offers are accepted (in order of lowest to highest) until there is excess demand. The contract sends the revised imbalance estimate to the network operator. The accepted non-critical users are paid at their offered price and the accepted offers are used to determine the energy price for the critical users – see final box in Fig. 1.

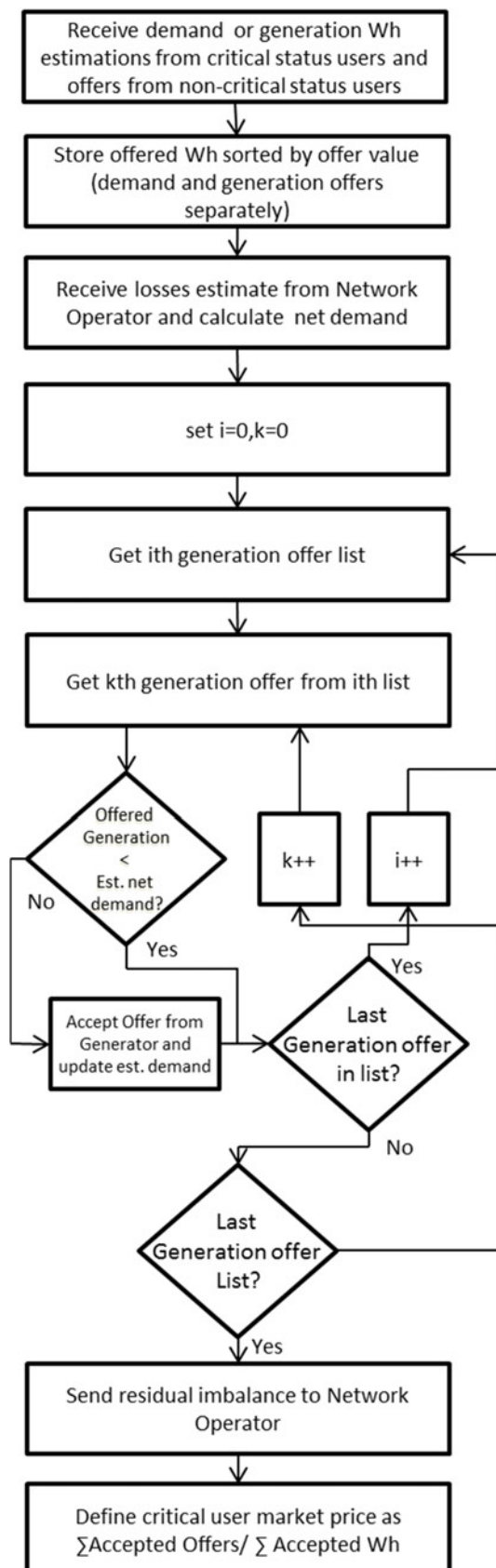


Fig. 1 Process for negotiation phase of supplier smart contract

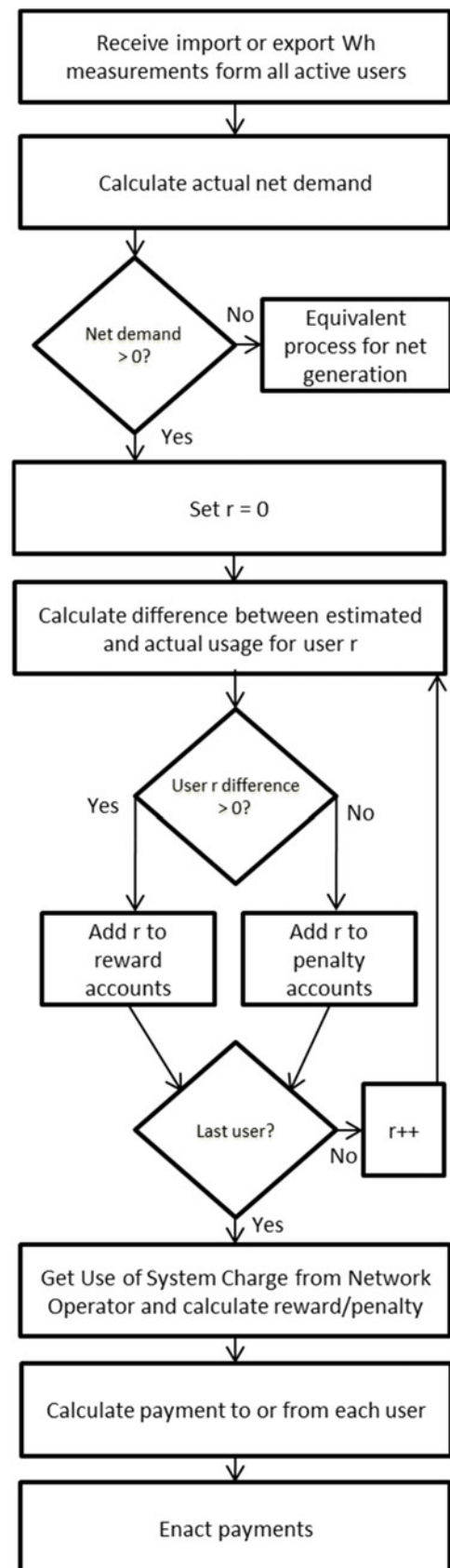


Fig. 2 Process for settlement phase of supplier smart contract

After usage, there is a settlement stage. All users submit their actual metered usage in Wh to the contract. Critical users pay the market price for their usage. Where users meet their estimate they are rewarded. Where they differ from

their estimated use in a way that helps system imbalance (e.g. where the system operator would have to call upon emergency generation reserve and the user had reduced demand) they are rewarded. Where they exacerbate system

imbalance they are penalised. The reward payment is calculated according to

$$R_{\text{user}} = \frac{H_{\text{user}} \times |E_{\text{User}} - A_{\text{User}}| \times \sum_{\text{users}}^u E_{\text{user}}}{\sum_{\text{users}}^u E_{\text{user}} + \sum_{\text{users}}^u A_{\text{user}}} \quad (1)$$

where R_{user} is the calculated reward for a given user, H_{user} is a historical performance factor between 0 and 1, E_{user} is a users' estimated usage, A_{user} is a users' actual usage and U is the set of all users.

After all of the rewards have been calculated, the sum of all rewards is used to attribute penalties to the relevant accounts

$$P_{\text{user}} = \frac{|E_{\text{User}} - A_{\text{User}}| \times \sum_{\text{users}}^u R_{\text{user}}}{\left| \sum_{\text{users}}^u E_{\text{user}} - \sum_{\text{users}}^u A_{\text{user}} \right|} \quad (2)$$

where P_{user} is the calculated penalty for a given user.

This formulation means users that adjust to balance demand-generation on the system are paid by those that unbalance it. The historical performance factor, H_{user} , reduces the potential for gaming of the formula through submission of inaccurate estimates.

The smart contract code was written in the solidity programming language [18]. The contract uses the default value token of the Ethereum platform, the Ether – ETH. To participate, users must maintain a deposit from which their usage payment is taken (or into which they are paid).

2.2 Supplier smart contract test case – negotiation

To test the example smart contract 1000 user accounts were generated with a private blockchain simulation created with testrpc. The first 900 of the users were partitioned into 800 low-voltage (LV) users, 90 medium-voltage (MV) and 10 high-voltage (HV). They were designated as critical and their estimated demand and generation usages were created using Numpy's [19], lognormal function with parameters shown in Table 1. The resulting distributions are shown in Fig. 3.

For the remaining 100 users, designated as non-critical, generation offers were made with lognormal function parameters shown in Table 2. The ETH values depicted are not tied to existing Great British Pound (GBP) to ETH exchange rates. The 50 LV generator offers and prices are shown in Fig. 4.

The estimates for the 900 critical users and the offers from the 100 non-critical users were submitted to the smart contract on the testrpc Ethereum blockchain and the price setting function was called using go-ethereum. Estimated losses of 200 MWh were submitted by a designated network operator account.

Table 1 Characteristics of smart contract critical user load and generation estimates

| Description | Number | Probability distribution function parameters (mean, sigma, multiplier) |
|---------------|--------|------------------------------------------------------------------------|
| LV loads | 400 | 1, 1, 1 |
| LV generators | 400 | 1, 1, -0.9 |
| MV loads | 45 | 1, 1, 1,000,000 |
| MV generators | 45 | 1, 1, -900,000 |
| HV loads | 5 | 1, 1, 100,000,000 |
| HV generators | 5 | 1, 1, -100,000,000 |

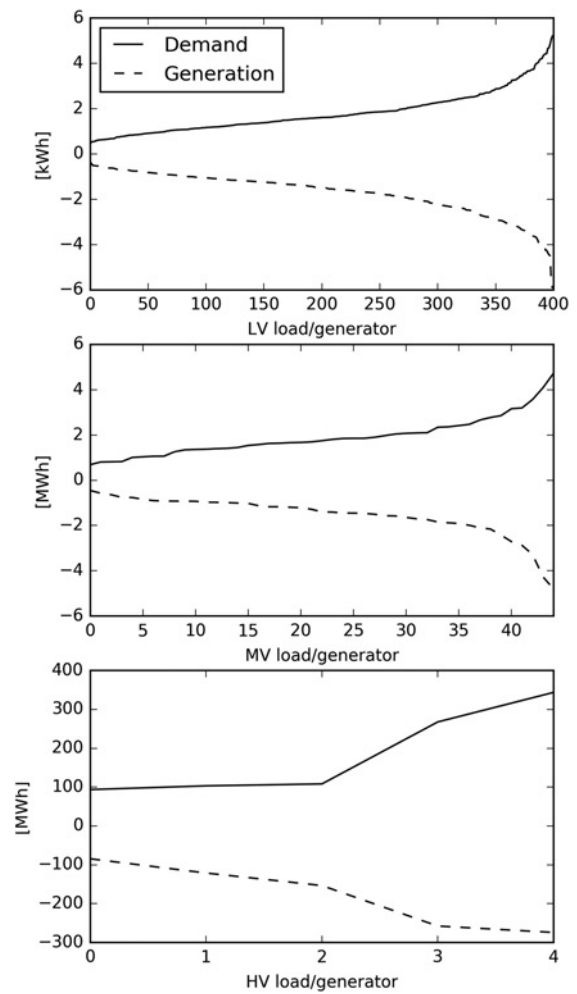


Fig. 3 Distributions of 400 LV load and 400 LV (top), 45 MV load and 45 MV generator (middle), five HV load and five HV generator estimates (bottom)

Table 2 Characteristics of smart contract participant load and generation – offers from non-critical users

| Description | No. | Offered price distribution function parameters (mean, sigma, multiplier) | Offered volume distribution function parameters (mean, sigma, multiplier) |
|---------------|-----|--------------------------------------------------------------------------|---------------------------------------------------------------------------|
| LV generators | 50 | 1, 1, 10 | 1, 1, -1 |
| MV generators | 40 | 1, 1, 10 | 1, 1, -1,000,000 |
| HV generators | 10 | 1, 1, 10 | 1, 1, -100,000,000 |

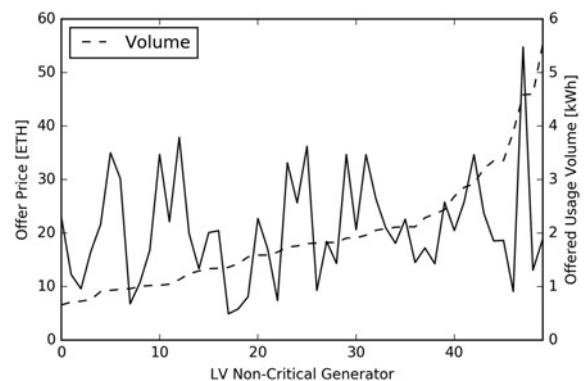


Fig. 4 Distribution of LV generation offers volume and price offers

Table 3 Adaptations of critical user LV and MV estimates to give simulated actual uses

| Description | Number | Multiplier function parameters (loc, scale) |
|---------------|--------|---------------------------------------------|
| LV loads | 400 | 1, 0.1 |
| LV generators | 400 | 1, 0.1 |
| MV loads | 45 | 1.01 |
| MV generators | 45 | 1.01 |

2.3 Supplier smart contract test case – settlement

Actual usage readings were simulated by multiplying the arrays of original estimates by normal distributions created by NumPy's normal function, with parameters as shown in Table 3. The HV generation and demand values were left unchanged as were all of the accepted non-critical users. Adjusted nominal losses of 190 MWh were submitted by a designated network operator account. The user accounts each submitted an actual usage reading to the smart contract. Those generators with accepted offers also submitted their usage values. Notional use of system (UoS) charges was set at 1 ETH per user.

3 Results

3.1 Negotiation

Once all estimates and offers were received by the smart contract, the market price setting function ran as described in Fig. 1. This resulted in the acceptance of 71 offers from the non-critical generators and a market price of 9 ETH/kWh for critical users.

3.2 Settlement

The settlement procedure ran as described in Fig. 2. The price paid per kWh for each user is shown in Fig. 5 (note the UoS charge is not included). In this test case as demand exceeded, generation in the negotiation phase, generation is paid (positive) and demand is charged. The variability in prices paid by the LV users is due to the increased spread of estimation error (Table 3) and the reward/penalty being attributed per Wh of error (Equations 1 and 2). The mean price received by LV generators is 7.1 ETH/kWh with a range from 10.9 to 2.0 ETH/kWh. The mean price paid by LV loads is 10.7 ETH/kWh with a range from 19.3 to 7.0 ETH/kWh. The mismatch between the load and generation prices is in part due to the losses (the notional UoS charge was excluded from these figures).

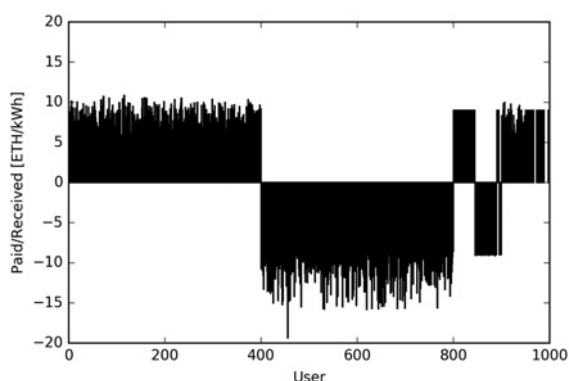


Fig. 5 Price per unit energy for each load/generator

4 Discussion

At present, owing to the proof of work consensus algorithm used by the Ethereum blockchain (and others), any computation performed by the blockchain-based smart contract is expensive compared to the same computation performed by a single computer. Whilst proposals to improve efficiency exist, and are planned, careful consideration must be given to what computation is done by a contract and what is done outside – whilst perhaps retaining a cryptographic root in the contract.

The next stage of this work is to create a set of tiered smart contracts, tied to the topology of the power network, such that there is clear competition for all services provided to customers including provision for system running costs (e.g. losses, stability, and protection), provision of import/export capacity as well as certainty of future availability (e.g. planning and maintenance) whilst not requiring active consumer engagement. The aim is to demonstrate a system that provides clear pricing signals to potential innovators and investors at all levels.

5 Conclusions

An energy supplier smart contract for automated negotiation, settlement and payment, with reward for system balancing support was demonstrated. This indicates potential for blockchain-based smart contracts to perform a supply role on the GB power system. The demonstrated smart contract includes reward for adjusting demand towards system balance with payments taken from those who adjusted away from system balance. In the example test case, a price was agreed, settled and payment enacted.

Energy supply companies can be viewed as competing sets of demand-generation negotiation rules. There is opportunity for suppliers to encode these rules as smart contracts. Consequently, there is a challenge to system regulators to arrange system information flows from metering (at all voltage levels) to smart contracts in such a way that efficacy of competition is maximised.

6 Acknowledgments

This research was conducted with the support of the EPSRC HubNet FlexiFund Project – Blockchain-based smart contracts for peer-to-peer energy trading using the GB smart metering system (EP/N030028/1) and the EU Horizon 2020 P2P smartTest project. The authors would like to acknowledge and thank the funders.

7 References

- Hough, D.: 'Competition and markets authority: energy market investigation', House of Commons Library, CBP7543, May 2016
- Szabo, N.: 'Smart contracts'. Available at <http://virtuallschool.edu/mon/Economics/SmartContracts.html>, accessed on 14 January 2017
- Szabo, N.: 'The god protocols'. Available at <https://web.archive.org/web/20160306022606/http://szabo.best.vwh.net/msc.html>, accessed on 14 January 2017
- Peters, G. W., Panayi, E., Chappelle, A.: 'Trends in crypto-currencies and blockchain technologies: a monetary theory and regulation perspective', SSRN 2646618, 2015
- Wood, G.: 'Ethereum: a secure decentralised generalised transaction ledger', Ethereum Proj. Yellow Pap., 2014
- 'Grid singularity'. Available at <http://gridsingularity.com/#/>
- 'Power ledger'. Available at <https://powerledger.io/>, accessed on 14 January 2017
- 'Ponton enerchain project'. Available at <http://enerchain.ponton.de/>, accessed on 14 January 2017
- 'Open energy exchange'. Available at <http://www.oex.org/en>, accessed on 14 January 2017
- 'VoltMarkets'. Available at <https://voltmarkets.com>, accessed on 14 January 2017
- 'SunExchange'. Available at <https://thesunexchange.com/>, accessed on 14 January 2017
- 'Co-tricity'. Available at <https://cotricity.com/en/about-co-tricity/>, accessed on 14 January 2017
- 'Electron'. Available at <http://www.electron.org.uk/>, accessed on 14 January 2017
- Zhumabekuly Aitzhan, N., Svetinovic, D.: 'Security and privacy in decentralized energy trading through multi-signatures, blockchain and anonymous messaging streams', *IEEE Trans. Dependable Secure Comput.*, 2016, **PP**, (99), pp. 1–1

- 15 Horta, J., Kofman, D., Menga, D.: 'Novel paradigms for advanced distribution grid energy management', 2016. Available at <http://www.theblockchain.com/docs/Novel%20paradigms%20for%20advanced%20distribution%20grid%20energy%20management.pdf>, accessed on 14 January 2017
- 16 'Brooklyn microgrid'. Available at <http://brooklynmicrogrid.com/>, accessed on 14 May 2017
- 17 'SolarCoin'. Available at <https://solarcoin.org/en/front-page/>, accessed on 17 January 2017
- 18 'Lee Thomas github profile'. Available at <https://github.com/4c656554>
- 19 Van Der Walt, S., Colbert, S. C., Varoquaux, G.: 'The NumPy array: a structure for efficient numerical computation', *Comput. Sci. Eng.*, 2011, **13**, (2), pp. 22–30